

REMARKS

This Amendment is filed in response to the Office Action mailed on February 27, 2007. All objections and rejections are respectfully traversed.

Claims 4-31 and 33-55 are currently pending.

Claims 55 is added.

Request for Interview

The Applicant respectfully requests a telephonic interview with the Examiner after the Examiner has had an opportunity to consider this Amendment, but before the issuance of the next Office Action. The Applicant may be reached at 617-951-3067.

Claim Rejection – 35 USC §101

At page 2 of the Office Action, claim 32 was rejected under 35 U.S.C. §101 because the claimed invention is directed to non-statutory subject matter.

The rejection is moot because Applicant has cancelled claim 32.

Claim Rejections – 35 USC § 102

At page 3 of the Office Action, claims 34-36, 38-41, 43 and 44 were rejected under 35 U.S.C. §102 as being anticipated by Hitz et al., U.S. Patent No. 6,721,764, hereinafter Hitz.

The present invention, as set forth in representative claim 34, comprise in part:

34. A method for generating a point-in-time restoration of a database to an active file system, comprising:
- storing a first snapshot, the first snapshot taken at a first time, the first snapshot including a set of database files;
 - storing a copy of a first log file, the copy of the first log file associated with the first snapshot, the copy of the first log file including information that had not yet been incorporated into the database files as of the first time;
 - storing a copy of a second log file, the copy of the second log file associated with a second snapshot taken at a second time subsequent to the first time, the copy of the second log file including information received subsequent to the first time that had not yet been incorporated into the database files as of the second time; and
- copying the first snapshot, the copy of the first log file, and the copy of the second log file to the active file system, to thereby restore at least a portion of the information received at the database subsequent to the first time without using the second snapshot.***

By way of background, Hitz discloses a method for keeping a file system in a consistent state and for creating read-only copies of a file system. The file system progresses from one self-consistent state to another self-consistent state. The set of self-consistent blocks on disk that is rooted by the root inode is referred to as a consistency point. To implement consistency points, new data is written to unallocated blocks on disk. A new consistency point occurs when the fsinfo block is updated by writing a new root inode for the inode file into it. Thus, as long as the root inode is not updated, the state of the file system represented on disk does not change. Snapshots are created, which are read-only copies of the file system. A snapshot uses no disk space when it is initially created. It is designed so that many different snapshots can be created for the same file system. Unlike prior art file systems that create a clone by duplicating the entire inode file and all of the indirect blocks,

the present invention duplicates only the inode that describes the inode file. A multi-bit free-block map file is used to prevent data from being overwritten on disk.

Applicant respectfully urges that Hitz does not disclose Applicant's claimed novel *copying the first snapshot, the copy of the first log file, and the copy of the second log file to the active file system, to thereby restore at least a portion of the information received at the database subsequent to the first time without using the second snapshot*. In further detail, in Applicant's claimed invention, a first snapshot is created at a first point in time. A first copy of the log file is also created, where the first log file includes changes to the database not included as of the first point in time. At a second point in time, a second snapshot is generated of the database. Additionally, a second copy of the log file is also created, where the second log file includes changes to the database not included in the database at the second point in time. When a user restores a snapshot to the active file system, the user may choose to use an older snapshot and any newer log files then the selected snapshot to restore most of the information.

In contrast, the Examiner states that Hitz discloses "copying the first snapshot, the copy of the first log file, and the copy of the second log file to the active file system" at col. 18, lines 45-60, which states:

"FIG. 18B is a diagram illustrating the creation of a snapshot. The snapshot is made for the entire file system 1830 by simply copying the inode 1810A of the inode file that is stored in fsinfo block 1810 into the snapshot inode 1822. By copying the inode 181A of the inode file, a new file of inodes is created representing the same file system as the active file system. Because the inode 1810A of the inode file itself is copied, no other blocks 1812-1820 need to be duplicated. The copied inode or snapshot inode 1822, is then copied into the inode file, which dirties a block in the inode file. For an inode file comprised of one or more levels of indirection, each indirect block is in

turn dirtied. This process of dirtying blocks propagates through all the levels of indirection. Each 4 KB block in the inode file on disk contains 32 inodes where each inode is 128 bytes long.”

In reference to the statement above, Hitz discloses copying the fsinfo block to create a snapshot. There is no disclosure in Hitz of restoring with two separate log files and a snapshot as claimed by Applicant.

Accordingly, Applicant respectfully urges that Hitz is legally insufficient to anticipate the present claims under 35 U.S.C. §102 because of the absence of the Applicant’s claimed novel *copying the first snapshot, the copy of the first log file, and the copy of the second log file to the active file system, to thereby restore at least a portion of the information received at the database subsequent to the first time without using the second snapshot.*

Claim Rejections – 35 USC § 103

At page 7 of the Office Action, claims 4-33, 42, and 52-54 were rejected under 35 U.S.C. § 103 as being unpatentable over Rudoff, US Patent No. 6,636,878, hereinafter Rudoff, in view of Hitz.

The present invention, as set forth in representative claim 4, comprise in part:

4. A method for generating a point-in-time restoration of a set of database files and a set of associated log files to an active file system, comprising:
 - selecting, by a user, a backup to restore therefrom, the backup comprising a snapshot of a file system including the set of database files and copies of the associated log files;
 - verifying the selected backup for coherency;
 - copying, in response to the backup being coherent, the snapshot of the set of database files to the active file system; and
 - copying, in response to the backup being coherent, the copies of the associated log files to the active file system.*

Rudoff discloses a mechanism for replicating and maintaining files in a space-efficient manner. A current file is replicated by associating the data blocks of the current file with both the current file and a new file.

Applicant respectfully urges that Rudoff and Hitz, taken alone or in combination do not teach or suggest Applicant's claimed novel *copying, in response to the backup being coherent, the copies of the associated log files to the active file system*. In further detail, in Applicant's claimed invention when a user selects a snapshot to restore, the associated log files are also restored. The log files include information that has not yet been incorporated in the database at the time the associated snapshot was created. The snapshot and the associated log file when restored return the database to the point in time that the snapshot and the copy of the log file were created.

The Examiner states that Rudoff does not disclose copying log files. Additionally, as stated above Hitz does not disclose copying the log files.

Hitz discloses a log at Col. 15, lines 1-10, which states:

“This is because only data corruption, which RAID protects against, or software can corrupt a WAFL file system. To avoid losing data when the system fails, WAFL may keep a non-volatile transaction log of all operations that have occurred since the most recent consistency point. This log is completely independent of the WAFL disk format and is required only to prevent operations from being lost during a system crash. However, it is not required to maintain consistency of the file system.” (Emphasis added)

In regard to the statement above, Hitz teaches away from using log file because it states that the log is not required to maintaining consistency of the file system. In Appli-

cant's claimed invention, the log file and the snapshot are used to restore the file system to certain point in time.

Accordingly, Applicant respectfully urges that Rudoff and Hitz, taken alone or in combination, are legally insufficient to make obvious the presently claimed invention under 35 U.S.C. § 103 because of the absence of the Applicant's claimed novel *copying, in response to the backup being coherent, the copies of the associated log files to the active file system*.

All independent claims are believed to be in condition for allowance.

All dependent claims are believed to be dependent from allowable independent claims.

Applicant respectfully solicits favorable action.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,

/Shannen C. Delaney/
Shannen C. Delaney
Reg. No. 51,605
CESARI AND MCKENNA, LLP
88 Black Falcon Avenue
Boston, MA 02210-2414
(617) 951-2500